

Goal: debug

Input: two tetrahedrons constructed by Zhiyu Ding as a test data.

The values and coordinates used to described the test data is showed below( picture 1 & 2 ).

1	5
2	2
3	0 0 0 33
4	0.5 0 0 20
5	0 0.5 0 11
6	0 0 0.5 30
7	0 0 -0.5 7
8	0 1 2 3
9	0 1 2 4

Picture 1, initial mesh data

1	9
2	2
3	0.25 0 0 16
4	0 0.25 0 2
5	0 0 0.25 5
6	0.25 0.25 0 14
7	0 0.25 0.25 7
8	0.25 0 0.25 18
9	0 0 -0.25 11
10	0.25 0 -0.25 8
11	0 0.25 -0.25 20
12	0 1 2 3 4 5
13	0 1 6 3 8 7

Picture 2, middle vertex data

Suppose that the old method with initial mesh data as input is the ground truth. After debug the part of data process, I got the results of both the original method( use initial mesh data ) and

new method( use both two parts of the data ). Table 1 is the contrast, black data means the result of ground truth and red ones are from the new method.

Ids	ids	tetNormal	tetNormal	vetPos	vetPos
0.125	1.0	0.5	0.7886	0.5	0.5
0.75	0.0	0.5	0.7886	0.5	0.5
0.25	0.0	0.0	0.7886	1.0	0.5
1.0	1.0	0.5	0.0	0.8846	1.0
0.0	0.0	0.0	0.5	0.5	1.0
0.0	0.0	0.5	0.5	1.0	0.5
1.0	1.0	0.0	0.5	0.5	0.5
0.0	0.0	0.5	0.0	0.1538	0.58
0.0	0.0	0.5	0.5	1.0	0.5
1.0	0.125	0.7886	0.5	0.5	1.0
0.0	0.75	0.7886	0.5	0.5	0.5
0.0	0.25	0.7886	0.0	0.5	0.29
0.125	1.0	0.5	0.7886	0.5	0.5
0.25	0.0	0.5	0.7886	0.5	0.5
0.25	0.0	1.0	0.2113	0.5	1.0
1.0	1.0	0.5	0.0	1.0	0.9
0.0	0.0	0.0	0.5	0.5	0.5
0.0	0.0	0.5	0.5	0.5	0.5
1.0	1.0	0.0	0.5	0.0	0.5
0.0	0.0	0.5	0.0	0.0	1.0
0.0	0.0	0.5	0.5	0.5	1.0
1.0	0.125	0.7886	0.5	1.0	0.5
0.0	0.25	0.7886	0.5	0.5	0.5
0.0	0.25	0.2113	1.0	0.1538	0.58
				1.0	0.5
				0.5	1.0
				0.5	0.5
				0.5	0.29
				0.5	0.5
				0.5	0.5
				0.5	0.0
				1.0	0.16

Table 1

Result: According to the data above, here comes a general result, namely both the two methods yield the same data result(For ids and tetNormals, they are both triples tuple. In ids, each component means its local ID in current cell, ID of current cell, ID of neighbor cell which shares the current face, respectively. Refers to tetNormals, they are three compnents of normal. As for

vetPos, they are x, y, z, scalar. ). In fact, vetPos are also the same, the difference in scalar value between them is caused by the normalize process which takes different normalize scale into account and these scale values are assigned in the data file, showed as picture 1 and 2.

Hence, the only difference is in which order that we save all these results. The new order here, is determined by the need of adding new data information, namely middle vertex information.

In order to obtain middle vertex data in a simple style, I made this “order” transformation.

Now, middle vertex data can be loaded and saved in their initial order. The reason why I add the transformation into these three arrays rather than add it into middle vertex data itself while keep the original part of code as before is that the later is far more complex.

Current version of program confirms the data produced by CPU code is correct, just the order is changed. It is clearly that I should modify the GPU code so that both parts are coincident. The current version of GPU is still the original one. It is the problem.

I will try to modify the shader next week.